UNIVERSIDAD DEL CEMA
Buenos Aires
Argentina

Serie
DOCUMENTOS DE TRABAJO

Área: Matemática, Economía, Negocios, Ingeniería

Serie de Machine Learning
Análisis de Componentes Principales (PCA)

Sergio A. Pernice

Diciembre 2020
Nro. 770

www.cema.edu.ar/publicaciones/doc_trabajo.html
UCEMA: Av. Córdoba 374, C1054AAP Buenos Aires, Argentina
ISSN 1668-4575 (impreso), ISSN 1668-4583 (en línea)
Editor: Jorge M. Streb; asistente editorial: Valeria Dowding jae@cema.edu.ar

Serie de Machine Learning
Análisis de Componentes Principales (PCA)

Sergio A. Pernice[1]

Universidad del CEMA
Av. Córdoba 374, Buenos Aires, 1054, Argentina

07 de diciembre de 2020

Abstract

En este documento presentamos la técnica de Principal Component Analysis (PCA). Es parte de la serie de documentos sobre machine learning. Es parte del contenido del curso "Métodos de Machine Learning para Economistas" de la Maestría en Economía de la UCEMA.

Keywords: Principal component analysis, Análisis de componentes principales, aprendizaje no supervisado.

1 Introducción: el problema de reducción dimensional (o compresión de datos)

1.1 Intuición

Consideremos la nube de puntos de la figura 1 si tuviéramos que encontrar la "mejor recta" que represente a esos puntos, cómo la dibujaría? Trate de dibujarla a mano antes de continuar.

Por supuesto que frente a una pregunta como esa la respuesta natural es "mejor con respecto a qué criterio? Ya vamos a ir a eso, pero apuesto a que su respuesta no fue muy diferente a la de la figura 2.

De la misma manera, consideremos la nube de puntos en 3-D en la figura 3 cuál sería en este caso la mejor recta? Ver figura 4. Y el mejor plano?

En general, nos podemos preguntar cuál es el mejor "k-plano", o subespacio afín (no necesariamente pasa por el origen) de k dimensiones, de un conjunto de puntos en d dimensiones ($k < d$).

Ahora tenemos que definir más claramente qué queremos decir por "mejor".

[1] sp@ucema.edu.ar
Los puntos de vista del autor no representan necesariamente la posición de la Universidad del CEMA.

Figure 1: Nube de puntos en el plano.

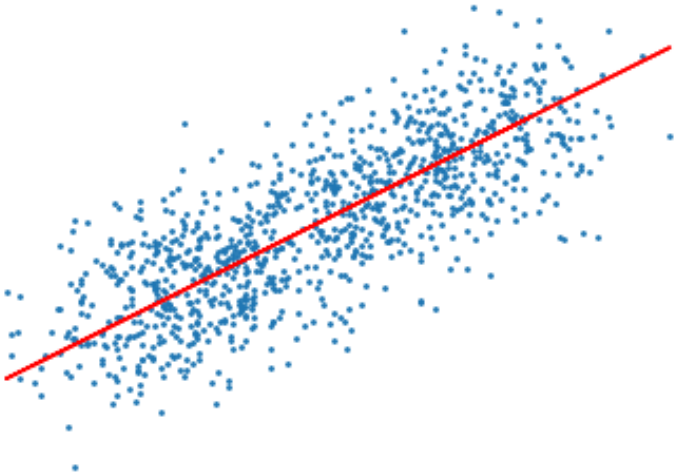


Figure 2: Nube de puntos en el plano con “mejor recta”.

1.2 El problema de la mejor recta entre dos puntos

Ver figura 5. Esa recta que minimiza la distancia es la mejor representación 1-D de puntos en 2-D. En el caso de la figura 5, con dos puntos, conocer la recta y conocer las coordenadas de los dos puntos en la recta, es equivalente a conocer las posiciones de los dos puntos.

Si son más de dos puntos, conocer la recta y conocer la coordenada de los dos puntos en la recta, *no* es equivalente a conocer la posición de los puntos, pero es la mejor aproximación 1-D a un problema intrínsecamente 2-D, como en la figura 2, intrínsecamente 3-D, como en la figura 4, o

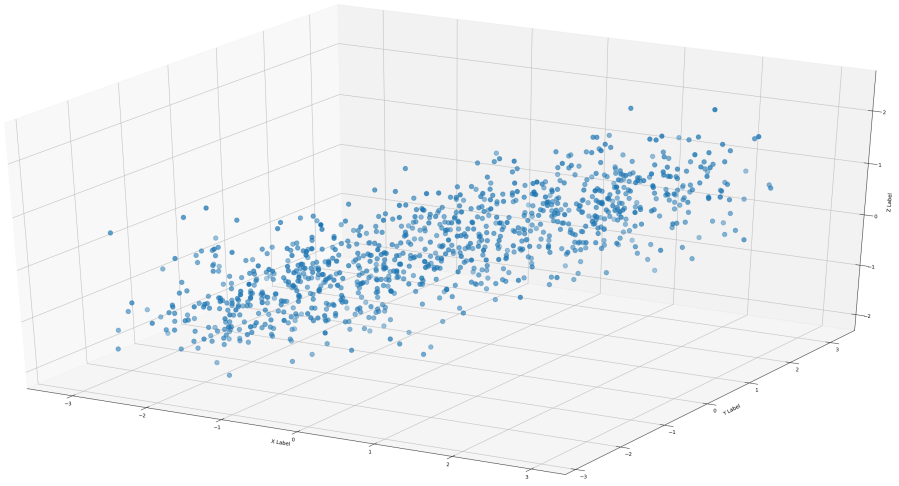


Figure 3: Nube de puntos en el espacio.

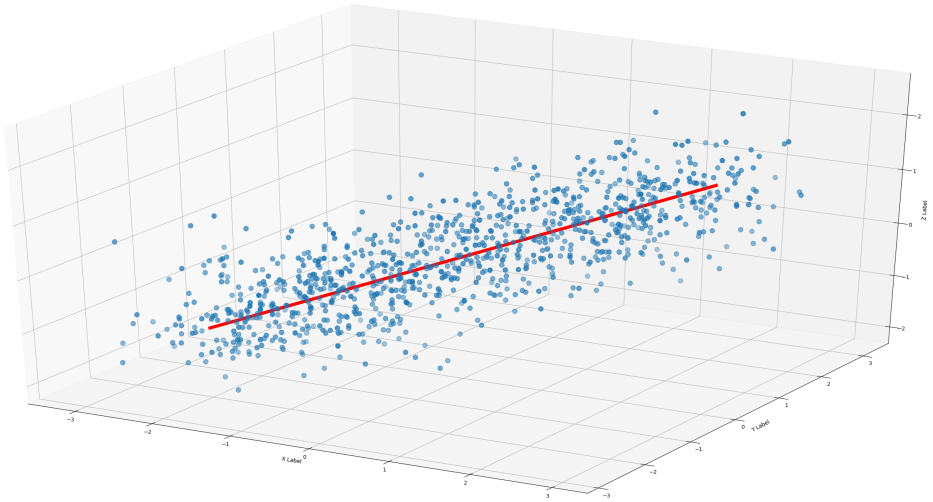
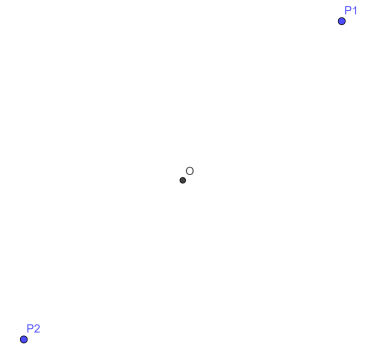


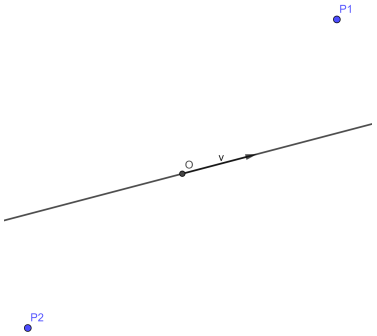
Figure 4: Nube de puntos en el espacio con “mejor recta”.

a un problema d-D.

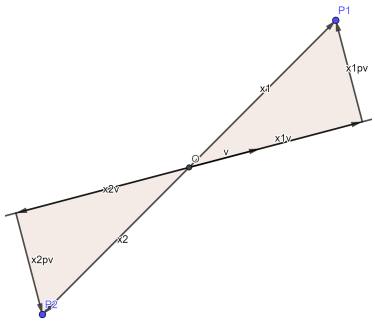
(a) Dos puntos en el plano.



(b) Nos paramos en el medio.



(c) Todo vector **v** de norma 1 define una recta.



(d) La distancia del origen a los puntos no depende de **v**.

Figure 5: La distancia al cuadrado del origen a los puntos no depende de **v**, pero el cuadrado de la distancia de los puntos a la recta sí depende de **v**. La mejor recta es la que minimiza la suma del cuadrado de las distancias de la recta a los puntos. Por Pitágoras, eso equivale a maximizar la suma del cuadrado de las proyecciones ortogonales de los puntos sobre la recta.

2 PCA, el problema geométrico: reducción dimensional

Tenemos n puntos en $\mathbb{R}^d$ correspondientes a los vectores $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$, $i = 0, \ldots, n-1$. El objetivo de PCA es encontrar aproximaciones de dichos vectores de la forma

$$\mathbf{x}_i \approx \bar{\mathbf{x}} + \sum_{j=1}^{k} a_{ij}\mathbf{v}_j, \quad i = 1, \ldots, n \tag{2.1}$$

con k vectores $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^n$, donde idealmente $k \ll d$.

Más específicamente, queremos encontrar el "k- plano" (hiperplano de k dimensiones que no necesariamente pase por el origen, también conocido como subespacio afín de k dimensiones) en $\mathbb{R}^d$, que minimice la distancia al cuadrado con los vectores $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$, $i = 0, \ldots, n-1$.

Para resolver el problema, tal como hicimos en la figura 5b para el problema de dos puntos, nos va a convenir trasladarnos al "centro" de los n puntos, ver figura 6. Esto se logra haciendo el

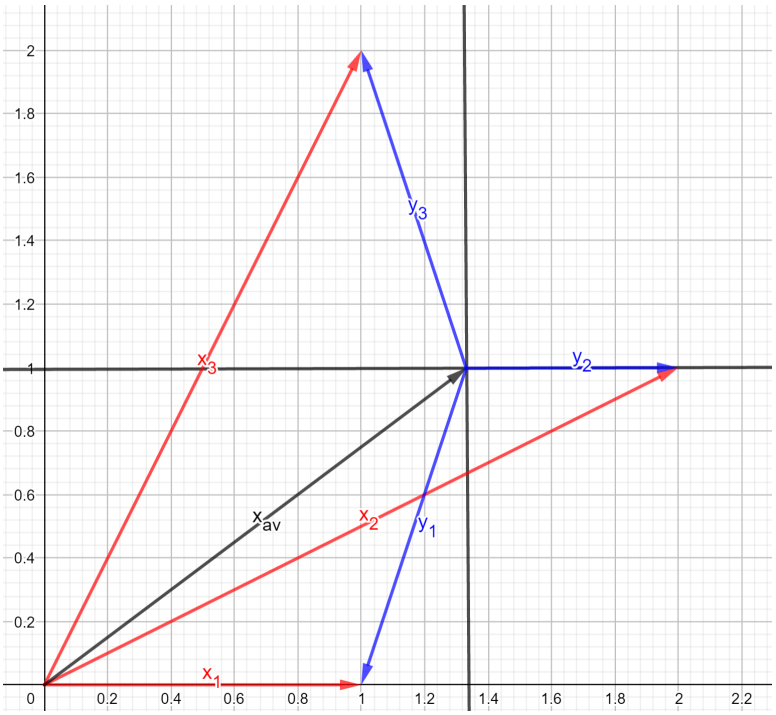


Figure 6: Nos trasladamos a un sistema de coordenadas en el centro de los datos.

cambio de variables:

$$\mathbf{y}_i = \mathbf{x}_i - \bar{\mathbf{x}}, \qquad i = 0, \ldots, n-1 \tag{2.2}$$

donde

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{x}_i, \quad \Rightarrow \quad \bar{\mathbf{y}} = \bar{\mathbf{x}} - \bar{\mathbf{x}} = \mathbf{0} \tag{2.3}$$

en este nuevo sistema de coordenadas no vamos a necesitar ordenada al origen, por lo que tratamos de encontrar aproximaciones de cada uno de los n vectores $\mathbf{y}_0, \ldots, \mathbf{y}_{n-1} \in \mathbb{R}^d$ como combinaciones lineales de k vectores $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^d$:

$$\mathbf{y}_i \approx \sum_{j=0}^{k-1} a_{ij} \mathbf{v}_j, \quad i = 0, \ldots, n-1 \tag{2.4}$$

Notar que si bien los vectores "viven" en $\mathbb{R}^d$, queremos aproximarlos en un k-plano, donde idealmente $k \ll d$.

2.1 $k = 1$, la mejor recta

Empecemos con $k = 1$. Planteamos la siguiente función objetivo:

$$\underset{\mathbf{v}:\|\mathbf{v}\|=1}{\operatorname{argmin}} \frac{1}{n} \sum_{i=0}^{n-1} \left((\text{distancia entre } \mathbf{y}_i \text{ y la línea generada por } \mathbf{v})^2 \right) \tag{2.5}$$

Dada una recta determinada por el vector $\mathbf{v}$ (de norma = 1), el vector $\mathbf{y}_i$ correspondiente a todo punto i se puede escribir como

$$\mathbf{y}_i = d_{\mathbf{v},i}\mathbf{v} + d_{\perp\mathbf{v},i}\mathbf{v}_{\perp,i} \tag{2.6}$$

donde $\mathbf{v}_{\perp,i}$ tiene norma 1 y es ortogonal a $\mathbf{v}$, de ahí la parte $\perp$ del subíndice. La parte i del subíndice es porque en más de dos dimensiones la dirección de $\mathbf{v}_{\perp,i}$, además de depender de $\mathbf{v}$, va a depender también de la posición $\mathbf{x}_i$ del punto i.

Como vimos en la figura 5d, el vector $\mathbf{y}_i$ es la hipotenusa del triángulo rectángulo cuyos catetos son $d_{\mathbf{v},i}\mathbf{v}$ y $d_{\perp\mathbf{v},i}\mathbf{v}_{\perp,i}$. El teorema de Pitágoras indica que para cada punto, $\|\mathbf{y}_i\|^2 = d^2_{\mathbf{v},i} + d^2_{\perp\mathbf{v},i}$. Por otro lado, para cada punto i, $\|\mathbf{y}_i\|^2$ está fijo. Por lo tanto

$$\sum_{i=0}^{n-1} \|\mathbf{y}_i\|^2 = \text{cte} = \sum_{i=0}^{n-1} \left(d^2_{\mathbf{v},i} + d^2_{\perp\mathbf{v},i} \right) = \sum_{i=0}^{n-1} d^2_{\mathbf{v},i} + \sum_{i=0}^{n-1} d^2_{\perp\mathbf{v},i} \tag{2.7}$$

La función objetivo (2.5) corresponde a minimizar el segundo término en la última expresión de (2.7). Pero como la suma de los dos términos es una constante, esto es equivalente a maximizar el primer término. Pero este es proporcional a la suma del cuadrado de las proyecciones ortogonales de los vectores $\mathbf{y}_i$ sobre $\mathbf{v}$. A dicha cantidad la vamos a llamar *varianza* de los datos en la dirección determinada por $\mathbf{v}$ (más sobre esto en sección 3):

$$\operatorname{Var}(\{\mathbf{y}_i\}, \mathbf{v}) = \frac{1}{n} \sum_{i=0}^{n-1} d^2_{\mathbf{v},i} \tag{2.8}$$

y el objetivo (2.5) se puede entonces plantear como

$$\underset{\mathbf{v}:\|\mathbf{v}\|=1}{\operatorname{argmax}} \left[\operatorname{Var}(\{\mathbf{y}_i\}, \mathbf{v}) \right] \tag{2.9}$$

Como vamos a ver en la sección 3, este modo de plantear el objetivo (2.5), tiene una interpretación estadística. Es encontrar la dirección que captura la máxima varianza de los datos.

Observando (2.6) vemos que $d_{\mathbf{v},i} = \mathbf{y}_i^\top \mathbf{v}$, por lo que con (2.8) el objetivo (2.9) es encontrar el vector unitario $\mathbf{v}$ que maximice la función:

$$\operatorname{Var}(\{\mathbf{y}_i\}, \mathbf{v}) = \frac{1}{n} \sum_{i=0}^{n-1} (\mathbf{y}_i^\top \mathbf{v})^2 = \frac{1}{n} \sum_{i=0}^{n-1} (\mathbf{v}^\top \mathbf{y}_i)(\mathbf{y}_i^\top \mathbf{v}) = \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{v}^\top (\mathbf{y}_i \mathbf{y}_i^\top) \mathbf{v} \tag{2.10}$$

Pero esto es una forma cuadrática asociada a la matriz simétrica $\in \mathbb{R}^{d\times d}$, semidefinida positiva:

$$A = \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{y}_i \mathbf{y}_i^\top, \qquad \operatorname{Var}(\{\mathbf{y}_i\}, \mathbf{v}) = \mathbf{v}^\top A \mathbf{v} \tag{2.11}$$

Reescribamos la matriz A en un formato conocido de nuestro estudio de regresiones. El i-ésimo vector $\mathbf{y}_i$ tiene componentes $y_{i,j}$, $j = 0, \ldots, d-1$:

$$\mathbf{y}_i = \begin{pmatrix} y_{i,0} \\ y_{i,1} \\ \vdots \\ y_{i,d-1} \end{pmatrix} \tag{2.12}$$

Construyamos la matriz:

$$\underset{n\times d}{D} = \begin{pmatrix} -\mathbf{y}_0^\top - \\ -\mathbf{y}_1^\top - \\ \vdots \\ -\mathbf{y}_{n-1}^\top - \end{pmatrix} = \begin{pmatrix} y_{0,0} & y_{0,1} & \cdots & y_{0,d-1} \\ y_{1,0} & y_{1,1} & \cdots & y_{1,d-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n-1,0} & y_{n-1,1} & \cdots & y_{n-1,d-1} \end{pmatrix} \tag{2.13}$$

es decir, cada fila de D corresponde a un punto de la nube de puntos que queremos aproximar. $y_{i,j}$ corresponde al valor de la $j-$ésima coordenada ($j = 0, \ldots, j-1$) del $i-$ésimo punto ($i = 0, \ldots, n-1$) en el sistema de referencia en el centro de la nube.

Vemos que la matriz de (2.11) es

$$A = \frac{1}{n} D^\top D \tag{2.14}$$

Entonces, el objetivo (2.9), equivalente al objetivo original (2.5), es:

$$\underset{\mathbf{v}:\|\mathbf{v}\|=1}{\operatorname{argmax}} \left[\operatorname{Var}\left(\{\mathbf{y}_i\}, \mathbf{v}\right)\right] = \underset{\mathbf{v}:\|\mathbf{v}\|=1}{\operatorname{argmax}} \left[\mathbf{v}^\top A \mathbf{v}\right] = \underset{\mathbf{v}:\|\mathbf{v}\|=1}{\operatorname{argmax}} \left[\frac{1}{n} \mathbf{v}^\top \left(D^\top D\right) \mathbf{v}\right] \tag{2.15}$$

Cuando analizamos las circunstancias en las que los problemas de regresión tienen solución y cuándo, si existe, dicha solución es única, nos enfrentamos con el análisis de una matriz con la misma estructura que A. Repetimos aquí dicho análisis: la matriz D en (2.13) es $\mathbb{R}^{n\times j}$, por lo que la matriz $A = (1/n) D^\top D \in \mathbb{R}^{d\times d}$ es simétrica $A^\top = A$ y semidefinida positiva.

Que es simétrica se ve así: $A^\top = \left(D^\top D\right)^\top = D^\top \left(D^\top\right)^\top = D^\top D = A$.

Una matriz cuadrada $A \in \mathbb{R}^{d\times d}$ es semidefinida positiva si para todo vector $\mathbf{v} \in \mathbb{R}^{d\times 1}$, $\mathbf{v}^\top A \mathbf{v} \geq 0$. La matriz A en (2.14) cumple con esta condición, ya que para todo vector $\mathbf{v}$, el cuadrado de la norma ℓ_2 del vector $\mathbf{z} = D\mathbf{v} \in \mathbb{R}^{n\times 1}$ es $\|\mathbf{z}\|_2^2 = \mathbf{z}^\top \mathbf{z} = \mathbf{v}^\top D^\top D \mathbf{v} = \mathbf{v}^\top A \mathbf{v}$. Como sabemos, $\|\mathbf{z}\|_2 \geq 0$, e $\|\mathbf{z}\|_2 = 0 \Leftrightarrow \mathbf{z} = \mathbf{0}$.

Sabemos que toda matriz simétrica de $d \times d$ es "diagonalizable" en la base de autovectores. Tiene d autovalores reales (contando cada uno tantas veces como sea su multiplicidad) y d autovectores reales ortonormales. Además, como es semidefinida positiva, sus autovalores son positivos o cero, pero nunca negativos.

Recordemos que una matriz simétrica se puede descomponer como una suma productos de sus autovectores ortonormales, multiplicados por el correspondiente autovalor:

$$A = \sum_{i=0}^{d-1} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top \tag{2.16}$$

Dado cualquier vector $\mathbf{w}$ de modulo 1,

$$\mathbf{w}^\top A\mathbf{w} = \sum_{i=0}^{d-1} \lambda_i \mathbf{w}^\top(\mathbf{v}_i\mathbf{v}_i^\top)\mathbf{w} = \sum_{i=0}^{d-1} \lambda_i(\mathbf{w}^\top\mathbf{v}_i)(\mathbf{v}_i^\top\mathbf{w}) = \sum_{i=0}^{d-1} \lambda_i(\mathbf{w}^\top\mathbf{v}_i)^2 \tag{2.17}$$

Como A es semidefinida positiva, sus autovalores son positivos o cero. Por otro lado $0 \leq (\mathbf{w}^\top\mathbf{v}_i)^2 \leq 1$, dado que ambos vectores tienen módulo unidad. Por lo tanto, (2.17) es un promedio ponderado de los autovalores de A. Dicho promedio obviamente se maximiza cuando $\mathbf{w} = \mathbf{v}_{\max}$, donde $\mathbf{v}_{\max}$ es el autovector de A con máximo autovalor.

El vector $\mathbf{v}$ que genera una recta que maximiza el objetivo (2.9) es entonces el autovector con máximo autovalor de la matriz $A = D^\top D$.✓

Ya identificamos la recta óptima, ahora queremos el valor de las coordenadas de los puntos en esta recta. Esto es simplemente la proyección ortogonal de los puntos sobre la recta. Si llamamos $\mathbf{v}_0$ al autovector identificado antes con norma 1 (Python automáticamente devuelve los autovectores normalizados a 1), entonces la coordenada del punto i-ésimo (2.12) en la recta óptima es:

$$c_{i,0} = \mathbf{y}_i^\top \mathbf{v}_0 \tag{2.18}$$

el nombre "c" viene de "coordenada", y el subíndice "$i, 0$" significa que es la proyección del punto i sobre el autovector vector 0 (correspondiente al máximo autovalor).

Para tener un vector de $\mathbb{R}^n$ que contenga todas las coordenadas de los n puntos $\mathbf{y}_i$, $i = 0, \ldots, n-1$, observando la matriz D en (2.13) vemos que tal vector es:

$$\underset{n\times 1}{\mathbf{c}_0} = \underset{n\times d}{D}\ \underset{d\times 1}{\mathbf{v}_0} \tag{2.19}$$

este vector es la mejor "reducción dimensional", o "compresión" de nuestros n puntos desde $\mathbb{R}^d$ a $\mathbb{R}$ (hay un solo número por punto, en vez de d como había originalmente).

Si queremos la posición de esta reducción dimensional de los puntos en el espacio original $\mathbb{R}^d$, volvemos a nuestro sistema de coordenadas original así:

$$\mathbf{x}_{i,\text{red1D}} = \bar{\mathbf{x}} + c_{i,0}\mathbf{v}_0 = \bar{\mathbf{x}} + (\mathbf{y}_i^\top \mathbf{v}_0)\,\mathbf{v}_0 \in \mathbb{R}^{d\times 1}, \quad i = 0, \ldots, n-1 \tag{2.20}$$

esta es la mejor aproximación 1-D de nuestros puntos en $\mathbb{R}^d$:

$$\mathbf{x}_{i,\text{red1D}} \approx \mathbf{x}_i \tag{2.21}$$

el símbolo $\approx$ no debe interpretarse como que el lado izquierdo es necesariamente una buena aproximación del lado derecho. En algunos problemas lo será y en otros no, dependiendo de la distribución de puntos $\mathbf{x}_i$. Pero es la mejor aproximación 1-D de nuestros puntos bajo el criterio de minimización de la distancia Euclidiana.

Si trasponemos la ecuación (2.20), $\mathbf{x}_{i,\text{red1D}}^\top = \bar{\mathbf{x}}^\top + c_{i,0}\mathbf{v}_0^\top \in \mathbb{R}^{1\times d}$. Teniendo en cuenta que el vector (2.19) contiene todos los $c_{i,0}$, y que la componente ij del producto outer entre dos vectores $\mathbf{u}\mathbf{w}^\top$ es $u_i w_j$, vemos que

$$X_{\text{red1D}} = \mathbf{one}_{n\times 1}\bar{\mathbf{x}}^\top + \mathbf{c}_0\mathbf{v}_0^\top = \mathbf{one}_{n\times 1}\bar{\mathbf{x}}^\top + (D\mathbf{v}_0)\,\mathbf{v}_0^\top = \mathbf{one}_{n\times 1}\bar{\mathbf{x}}^\top + D\,(\mathbf{v}_0\mathbf{v}_0^\top) \tag{2.22}$$

donde $\mathbf{one}_{n\times 1}$ es un vector de $n \times 1$ con todos los elementos iguales a 1[2]. La fila i-ésima de la matrix X_{red1D} (de $n \times d$) son las coordenadas en $\mathbb{R}^d$ de la reducción 1-D del i-ésimo punto.

Notemos que en el segundo término de la última expresión de (2.22) aparece la matriz $\mathbf{v}_0\mathbf{v}_0^\top$, que es el proyector sobre el subespacio spaneado por $\mathbf{v}_0$. El hecho de multiplicar a dicho proyector por D por izquierda, significa que estamos proyectando las filas de D sobre dicho subespacio. Pero las filas de D son los vectores de nuestros puntos en el sistema de referencia en el centro de los puntos, ver (2.13). Es decir que las filas del segundo término de la última expresión de (2.22) son las proyecciones de los vectores de nuestros puntos en el subespacio spaneado por $\mathbf{v}_0$.

El primer término de la última expresión de (2.22) es una matriz cuyas filas son toda iguales y corresponden a la posición del centro de la nube de puntos en el sistema de referencia original. Entonces las n filas de X_{red1D} son los vectores de $\mathbb{R}^d$ correspondientes a la proyección de los n puntos. Con entrenamiento, una ecuación como (2.22) se puede escribir directamente sin ninguna derivación ya que (con entrenamiento) resulta "obvia".

2.2 El mejor k-plano

El análisis anterior trivialmente se extiende al mejor k-plano, y la solución es que dicho k-plano, en el sistema en el centro de la nube, es el span de los k autovectores con los k mayores autovalores.

La ecuación (2.18) es ahora

$$c_{i,j} = \mathbf{y}_i^\top \mathbf{v}_j, \quad j = 0, \ldots, k-1 \tag{2.23}$$

donde $c_{i,j}$ es la coordenada del i-ésimo punto en el subespacio spaneado por el j-ésimo autovector (el autovector cuyo autovalor es el j-ésimo en tamaño). (2.19) es

$$\underset{n\times 1}{\mathbf{c}_j} = \underset{n\times d}{D}\ \underset{d\times 1}{\mathbf{v}_j}, \quad j = 0, \ldots, k-1 \tag{2.24}$$

La matriz (2.22) generaliza a

$$X_{\text{redkD}} = \mathbf{one}_{n\times 1}\bar{\mathbf{x}}^\top + D\left(\sum_{j=0}^{k-1} \mathbf{v}_j\mathbf{v}_j^\top\right) \tag{2.25}$$

El proyector sobre el "mejor" k-plano es la suma de los proyectores sobre los 1-planos (o rectas) de cada autovector porque estos son ortogonales entre sí.

En Python, dada la matriz $A = (1/n)D^\top D$ en (2.14), la función **eigh** de la library linalg de numpy nos da la matriz:

$$U_k = \begin{pmatrix} | & | & & | & | \\ \mathbf{v}_{k-1} & \mathbf{v}_{k-2} & \cdots & \mathbf{v}_1 & \mathbf{v}_0 \\ | & | & & | & | \end{pmatrix} \tag{2.26}$$

[2]En numpy este vector es np.ones(n,1), pero en realidad no es necesario hacer la operación $\mathbf{one}_{n\times 1}\bar{\mathbf{x}}^\top$ explícitamente en Python porque $\bar{\mathbf{x}}^\top$, de "shape" $(1, d)$, es "broadcasted" automáticamente a $\mathbf{one}_{n\times 1}\bar{\mathbf{x}}^\top$, de shape (n, d). en (2.22).

donde la última columna es el "mayor" autovector, la anteúltima el siguiente, etc. Con U_k, por la manera "conjunto de productos matriz-vector" de ver el producto de matrices, los vectores proyección de los n puntos sobre los k autovectores (2.24) so pueden ordenar como las columnas de la matriz

$$\underset{n\times k}{C} = \underset{n\times d}{D}\ \underset{d\times k}{U_k} \tag{2.27}$$

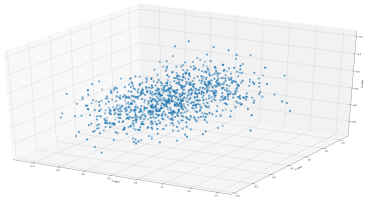
que se pueden graficar para $k = 1, 2$ o 3 y son la mejor representación en k dimensiones de nuestros puntos de $\mathbb{R}^d$.

Por la manera "producto outer" de ver el producto de matrices, (2.25) se puede escribir de manera compacta así:

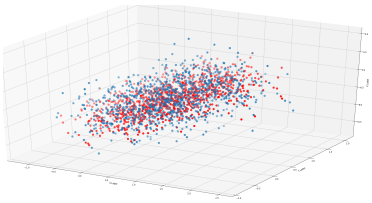
$$X_{\text{redkD}} = \mathbf{one}_{n\times 1}\bar{\mathbf{x}}^\top + DU_kU_k^\top \tag{2.28}$$

que corresponden al mejor k-plano en $\mathbb{R}^d$.

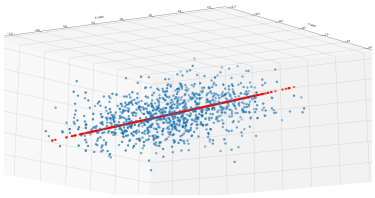
En la figura 7 vemos en acción a las ecuaciones (2.27) y (2.28) para una proyección 2-D de puntos en 3-D.



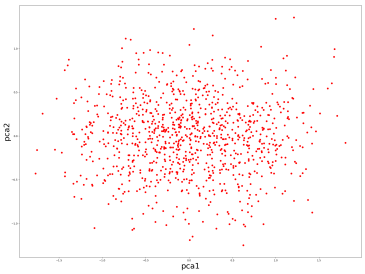
(a) Nube original de puntos 3-D.



(b) Los puntos rojos son la proyección de los puntos de 7a al mejor plano (2-D), generados con la ecuación (2.28).



(c) Desde esta perspectiva se percibe que los puntos rojos de 7b yacen en un plano.



(d) La proyección 2-D de las figuras 7b y 7c, sin el espacio 3-D en el que está inmerso, generado con la ecuación (2.27).

Figure 7: Reducción dimensional de 3-D a 2-D.

2.3 Pre-procesamiento de datos

Pensemos en una tabla con una gran cantidad de datos sobre todos los países: PBI, PBI per cápita, nivel de desarrollo humano, índice de Gini, etc. Para esos datos, n = número de países, y d = número de tipos de datos que hay en la tabla. Pero los diferentes datos se miden en unidades completamente diferentes, cómo los hacemos comparables? En un primer intento, los datos de cada dimensión se dividen por la desviación estándar de dichos datos de esa dimension.

O supongamos que en una dimensión tenemos datos medidos en centímetros y en otra tenemos datos medidos en kilómetros, dividiendo cada dato por la desviación estándar se vuelven comparables.

Pero hay que tener cuidado, ya que aplicando estos procedimientos a ciegas, corremos el riesgo de eliminar varianzas genuinas de los datos que con este pre-procesamiento parcialmente se pierden. Por eso el pre-procesamiento de los mismos, al menos en el contexto del uso de PCA para ciencia de datos, requiere de cierto conocimiento previo de los mismos y es parte ciencia parte arte.

En un contexto puro de machine learning no supervisado, hay en principio mecanismos para que la máquina encuentre el mejor pre-procesamiento de los datos, más sobre esto más adelante en el curso.

3 Conexión con estadística

Asumamos que $x_0, x_1 \ldots, x_{d-1}$ son d variables aleatorias que ordenamos en un vector $\mathbf{x} \in \mathbb{R}^{d\times 1}$. Hay subyacente una distribución de probabilidades multivariada $p(\mathbf{x})$ que suponemos que no conocemos. Empíricamente obtenemos n samples de esta variable aleatoria vectorial. Ordenamos nuestra data así:

$$\underset{n\times j}{D_{\text{or}}} = \begin{pmatrix} -\mathbf{x}_0^\top - \\ -\mathbf{x}_1^\top - \\ \vdots \\ -\mathbf{x}_{n-1}^\top - \end{pmatrix} = \begin{pmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,d-1} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,d-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1,0} & x_{n-1,1} & \cdots & x_{n-1,d-1} \end{pmatrix} \tag{3.1}$$

el subíndice "or" refiere a que son los datos empíricos originales. Cada fila de D_{or} corresponde a un sample de nuestra variable aleatoria. $x_{i,j}$ corresponde al valor de la $j-$ésima variable aleatoria ($j = 0, \ldots, j-1$) del $i-$ésimo sample ($i = 0, \ldots, n-1$).

La estimación empírica de la media de la variable aleatoria j es:

$$\bar{x}_j = \frac{1}{n}\sum_{i=0}^{n-1} x_{i,j} \tag{3.2}$$

podemos vectorizar esta expresión, capturando en una misma ecuación vectorial la media de

todas nuestras variables aleatorias:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{x}_i \tag{3.3}$$

La estimación empírica de la varianza de la variable aleatoria j es:

$$\text{Var}\left(x_j\right) = \frac{1}{n-1}\sum_{i=0}^{n-1}\left(x_{i,j} - \bar{x}_j\right)^2 \tag{3.4}$$

y la estimación empírica de la covarianza entre la variable aleatoria j y la k es:

$$\text{Cov}\left(x_j, x_k\right) = \frac{1}{n-1}\sum_{i=0}^{n-1}(x_{i,j} - \bar{x}_j)(x_{i,k} - \bar{x}_k) \tag{3.5}$$

el $n-1$ en el denominador de (3.4) y (3.5) hace a esos estimadores "unbiased". De todos modos nosotros vamos a estar trabajando con muchos datos, es decir, n lo suficientemente grande como para que $1/(n-1) \approx 1/n$, por lo que de aquí en más pasamos a reemplazar ese denominador por $1/n$.

Mirando las ecuaciones (3.2-3.5), es natural "trasladarnos", en el espacio de nuestras variables, de modo que el origen coincida con $\bar{\mathbf{x}}$ y la media en las nuevas variables sea cero. Es decir, haciendo el cambio de variables

$$\mathbf{y} = \mathbf{x} - \bar{\mathbf{x}}, \quad \Rightarrow \quad \bar{\mathbf{y}} = \bar{\mathbf{x}} - \bar{\mathbf{x}} = \mathbf{0} \tag{3.6}$$

tal como hicimos en la figura 2.

En estas nuevas variables, ordenamos nuestros samples como en la matriz D_{or} de (3.1), pero sin el subíndice "or":

$$\underset{n\times j}{D} = \begin{pmatrix} -\mathbf{y}_0^\top - \\ -\mathbf{y}_1^\top - \\ \vdots \\ -\mathbf{y}_{n-1}^\top - \end{pmatrix} = \begin{pmatrix} y_{0,0} & y_{0,1} & \cdots & y_{0,d-1} \\ y_{1,0} & y_{1,1} & \cdots & y_{1,d-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n-1,0} & y_{n-1,1} & \cdots & y_{n-1,d-1} \end{pmatrix} \tag{3.7}$$

Notemos que esta matriz es idéntica a (2.13) si identificamos a los samples de nuestras variables aleatorias con las coordenadas de los puntos que estudiamos en la sección 2.

En las nuevas variables, las estimaciones empíricas de las varianzas y covarianzas toman la forma

$$\text{Var}\left(y_j\right) = \frac{1}{n}\sum_{i=0}^{n-1} y_{i,j}^2 \tag{3.8}$$

$$\text{Cov}\left(y_j, y_k\right) = \frac{1}{n}\sum_{i=0}^{n-1} y_{i,j}y_{i,k} \tag{3.9}$$

donde recordamos que simplificamos el denominador de $n-1$ a n como fue explicado antes.

Teniendo en cuenta que el elemento ik de la matriz D en (3.7) es $y_{i,k}$ y que el elemento ji de la matriz transpuesta $D^\top$ es $y_{i,j}$, reconocemos en (3.8-3.9) los elementos del producto de matrices $D^\top$ por D, por lo que llegamos a la siguiente expresión vectorizada:

$$\underset{d\times d}{\Sigma} = \frac{1}{n}\underset{d\times n}{D}^{\top}\underset{n\times d}{D} \tag{3.10}$$

Los elementos diagonales de Σ son las varianzas (3.8) de las variables y_i, y los elementos no diagonales son las respectivas covarianzas (3.9).

Notamos que la matriz varianza-covarianza (3.10) es exactamente igual a la matriz (2.14), cuyos máximos autovectores era la solución de nuestro problema!

Con la identificación mencionada antes entre los samples de nuestras variables aleatorias con las coordenadas de los puntos que estudiamos en la sección 2 vemos que la solución de PCA coincide con encontrar la dirección en el espacio de nuestras variables aleatorias que maximizan la varianza en el sentido estadístico.

4 PCA vs. regresiones

En la figura 8 comparamos, para $k = 1$, el problema al que nos enfrentamos cuando hacemos regresiones (en azul) con el problema planteado aquí (en rojo).

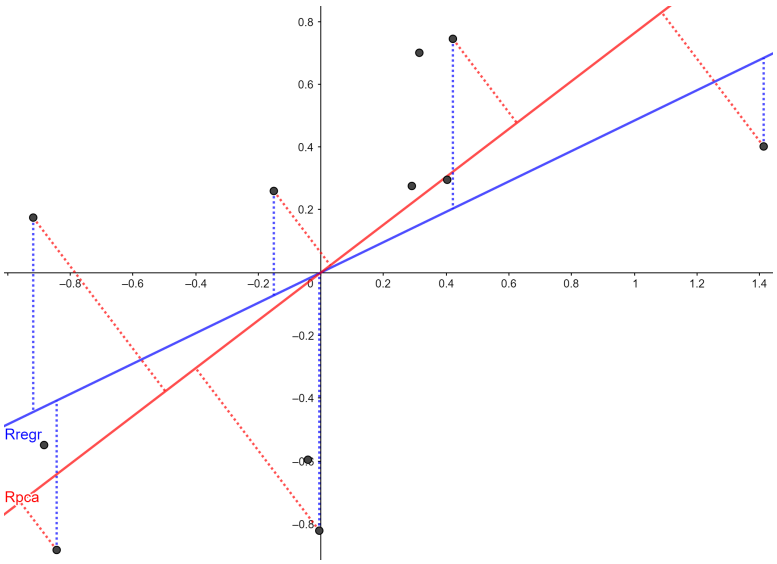


Figure 8: Mismos datos, diferentes rectas óptimas con PCA y regresiones. Las líneas punteadas rojas muestran lo que minimiza PCA y las azules lo que minimizan las regresiones.

En regresiones hay una variable que se diferencia de las otras, sirve para problemas en los que tenemos razones para suponer que las variables dependientes (la dimension horizontal en la figura 8) "explican" a la variable independiente (la dimension vertical en la figura 8). Por lo

tanto queremos minimizar el cuadrado del error de la predicción de nuestro modelo (recta azul). En la figura 8, minimizamos la suma de los cuadrados de las longitudes de los segmentos azules punteados. Notar que dichos segmentos son verticales, y su longitud mide el error de predicción para valores dados de la variable independiente.

Por el contrario, en PCA no hacemos diferenciación entre variables explicativas y variables a ser explicadas. Todas son equivalentes a priori, y dejamos que los datos nos indiquen si existen o no unas pocas variables cuyo efecto es suficiente para aproximar con precisión los datos empíricos. La respuesta por sí o por no es objetiva, y depende, como vimos, de la magnitud relativa de los autovalores de la matriz varianza-covarianza.

Sin embargo, en la práctica, el uso productivo de PCA requiere de un pre-procesamiento de los datos, que a su vez típicamente presupone cierto conocimientos parcial de los mismos, ver sección 2.3.

5 Conclusiones

PCA es una delas metodologias mas poderosas de reduccion dimensional (lineal). El método asume que la variación en los datos corresponde a información interesante. Sin embargo es fácil imaginar escenarios en los que dicha variación corresponde a ruido en lugar de señal, en cuyo caso la PCA puede no producir resultados esclarecedores.